Ag. Stat. 1.1 Agricultural Informatics (2+1=3)

Theory

1. Introduction to Computers,
2. Anatomy of Computers,
3. Memory Concepts, Units of Memory,
4. Operating System, definition and types,
5. Applications of MS-Office for creating, Editing and Formatting a document,
6. Data presentation, tabulation and graph creation, statistical analysis, mathematical expressions,
7. Database, concepts and types, creating database, uses of DBMS in Agriculture,
8. Internet and World Wide Web (WWW), Concepts and components.
9. e-Agriculture, concepts, design and development.
10. Application of innovative ways to use information and communication technologies (IT) in Agriculture.
12. IT application for computation of water and nutrient requirement of crops,
13. Computer-controlled devices (automated systems) for Agri-input management, Smartphone mobile apps in Agriculture for farm advises, market price, postharvest management etc;
14. Geospatial technology, concepts, techniques, components and uses for generating valuable agri-information.
15. Decision support systems, concepts, components and applications in Agriculture,
17. Preparation of contingent crop-planning and crop calendars using IT tools.

Practical

1. Study of Computer Components, accessories, practice of important DOS Commands.
2. Introduction of different operating systems such as windows, Unix/ Linux, Creating, Files & Folders, File Management.
3. Use of MS-WORD and MS Power-point for creating, editing and presenting a scientific Document.
4. MS-EXCEL - Creating a spreadsheet, use of statistical tools, writing expressions, creating graphs, analysis of scientific data, handling macros.
5. MS-ACCESS: Creating Database, preparing queries and reports, demonstration of Agri-information system.
7. Preparation of Inputs file for CSM and study of model outputs, computation of water and nutrient requirements of crop using CSM and IT tools.
8. Use of smart phones and other devices in agro-advisory and dissemination of market information.
9. Introduction of Geospatial Technology, for generating information important for Agriculture.
Basics of Computer

Computer
A computer is an electronic device, operating under the control of instructions stored in its own memory that can accept data (input), process the data according to specified rules, produce information (output), and store the information for future use.

Functionalities of a computer (Data processing cycle)
Any digital computer carries out five functions in gross terms:
1. Takes data as input
2. Store the data in memory and use them when required
3. Process the data and convert in useful instruction
4. Generate the output
5. Control all of above steps.

![](image)

Computer Components
Any kind of computers consists of HARDWARE AND SOFTWARE

Hardware:
Computer hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to the physical parts or components of a computer such as the monitor, mouse, keyboard, computer data storage, hard drive disk (HDD), system unit (graphic cards, sound cards, memory, motherboard and chips), etc. all of which are physical objects that can be touched

Input Devices
Input device is any peripheral (piece of computer hardware equipment to provide data and control signals to an information processing system such as a computer or other information appliance.
Input device Translate data from form that humans understand to one that the computer can work with. Most common are keyboard and mouse
Central Processing Unit (CPU)
A CPU is brain of a computer. It is responsible for all functions and processes. Regarding computing power, the CPU is the most important element of a computer system.
The CPU is comprised of three main parts:

1. **Arithmetic Logic Unit (ALU)**:
   - Executes all arithmetic and logical operations. Arithmetic calculations like addition, subtraction, multiplication and division. Logical operation like compare numbers, letters, or special characters.

2. **Control Unit (CU)**: controls and co-ordinates computer components.
   - 1. Read the code for the next instruction to be executed.
   - 2. Increment the program counter so it points to the next instruction.
   - 3. Read whatever data the instruction requires from cells in memory.
   - 4. Provide the necessary data to an ALU or register.
   - 5. If the instruction requires an ALU or specialized hardware to complete, instruct the hardware to perform the requested operation.
3. **Registers**: Stores the data that is to be executed next, "very fast storage area".

![Diagram of computer processing unit]

***Just For Note : Instruction cycle -----------cycle that the cpu follows from boot-up until the computer has shut down in order to process instruction. It contain following stage : 1.fetch stage 2. Decode stage 3.Execute stage***

**Output devices**-
An output device is any piece of computer hardware equipment used to communicate the results of data processing carried out by an information processing system (such as a computer) which converts the electronically generated information into human-readable form.

**Primary Memory:**

**RAM**: Random Access Memory (RAM) is a memory scheme within the computer system responsible for storing data on a temporary basis, so that it can be promptly accessed by the processor as and when needed. It is volatile in nature, which means that data will be erased once supply to the storage device is turned off. RAM stores data randomly and the processor accesses these data randomly from the RAM storage.

RAM is considered "random access" because you can access any memory cell directly if you know the row and column that intersect at that cell.

**ROM**: Read Only Memory. ROM is a permanent form of storage. ROM stays active regardless of whether power supply to it is turned on or off. ROM devices do not allow data stored on them to be modified.

**Secondary Memory**:
Stores data and programs permanently: its retained after the power is turned off

- **Hard drive (HD)**: A hard disk is part of a unit, often called a "disk drive," "hard drive," or "hard disk drive," that store and provides relatively quick access
to large amounts of data on an electromagnetically charged surface or set of surfaces.

- Optical Disk: an optical disc drive (ODD) is a disk drive that uses laser light as part of the process of reading or writing data to or from optical discs. Some drives can only read from discs, but recent drives are commonly both readers and recorders, also called burners or writers. Compact discs, DVDs, and Blu-ray discs are common types of optical media which can be read and recorded by such drives. Optical drive is the generic name; drives are usually described as "CD" "DVD", or "Bluray", followed by "drive", "writer", etc. There are three main types of optical media: CD, DVD, and Blu-ray disc. CDs can store up to 700 megabytes (MB) of data and DVDs can store up to 8.4 GB of data. Blu-ray discs, which are the newest type of optical media, can store up to 50 GB of data. This storage capacity is a clear advantage over the floppy disk storage media (a magnetic media), which only has a capacity of 1.44 MB.

**Flash Disk**

- A storage module made of flash memory chips. A Flash disks have no mechanical platters or access arms, but the term "disk" is used because the data are accessed as if they were on a hard drive. The disk storage structure is emulated.

<table>
<thead>
<tr>
<th>RAM</th>
<th>Hard Disk (Hard Drive)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Memory</strong></td>
<td>Storage</td>
</tr>
<tr>
<td>Smaller amount</td>
<td>Much larger amount</td>
</tr>
<tr>
<td>(typically 500 MB-6 GB)</td>
<td>(typically 80GB to 1000 GB)</td>
</tr>
<tr>
<td>Temporary storage of files</td>
<td>Permanent storage of files and programs</td>
</tr>
<tr>
<td>and programs</td>
<td></td>
</tr>
<tr>
<td>A little like your real</td>
<td>Like a file cabinet - has long-term</td>
</tr>
<tr>
<td>desktop - has only your</td>
<td>storage of work (it's safe from spills!)</td>
</tr>
<tr>
<td>current work on it (which</td>
<td></td>
</tr>
<tr>
<td>could be ruined by a</td>
<td></td>
</tr>
<tr>
<td>spill of Coke or coffee!</td>
<td></td>
</tr>
<tr>
<td>Contents disappear when</td>
<td>Contents remain when you turn off the</td>
</tr>
<tr>
<td>you turn off power to</td>
<td>power to the computer (they don't</td>
</tr>
<tr>
<td>the computer and when the</td>
<td>disappear unless you purposely delete</td>
</tr>
<tr>
<td>computer crashes</td>
<td>them), and when the computer crashes</td>
</tr>
<tr>
<td>Consists of chips</td>
<td>Consists of hard disks (platters)</td>
</tr>
<tr>
<td>(microprocessors)</td>
<td></td>
</tr>
<tr>
<td>When you want to use a</td>
<td>Holds the original copy of the program</td>
</tr>
<tr>
<td>program, a temporary</td>
<td>permanently</td>
</tr>
<tr>
<td>copy is put into RAM and</td>
<td></td>
</tr>
<tr>
<td>that's the copy you use</td>
<td></td>
</tr>
</tbody>
</table>

**Software**

Software is a generic term for organized collections of computer data and instructions, often broken into two major categories: system software that provides the basic non-task-specific functions of the computer, and application software which is used by users to accomplish specific tasks.

**Software Types**

A. System software
- System software is responsible for controlling, integrating, and managing the individual hardware components of a computer system so that other software and the users of the system see it as a functional unit without having to be concerned with the low-level details such as transferring data from memory to disk, or rendering text onto a display.

- Generally, system software consists of an operating system and some fundamental utilities such as disk formatters, file managers, display managers, text editors, user authentication (login) and management tools, and networking and device control software

B. Application software

- Application software is used to accomplish specific tasks other than just running the computer system.

- Application software may consist of a single program, such as an image viewer; a small collection of programs (often called a software package) that work closely together to accomplish a task, such as a spreadsheet or text processing system; a larger collection (often called a software suite) of related but independent programs and packages that have a common user interface or shared data format, such as Microsoft Office, which consists of closely integrated word processor, spreadsheet, database, etc.; or a software system, such as a database management system, which is a collection of fundamental programs that may provide some service to a variety of other independent applications

Unit of Measurements

Storage measurements: The basic unit used in computer data storage is called a bit (binary digit). Computers use these little bits, which are composed of ones and zeros, to do things and talk to other computers. All your files, for instance, are kept in the computer as binary files and translated into words and pictures by the software (which is also ones and zeros). This two number system, is called a “binary number system” since it has only two numbers in it. The decimal number system in contrast has ten unique digits, zero through nine.

1 byte = 8 bit
1KB = 1024 Byte
1MB = 1024 KB
1 GB = 1024 MB
1TB = 1024 GB

(1024 =2^{10} Byte)

Speed measurement: The speed of Central Processing Unit (CPU) is measured by Hertz (Hz), which represent a CPU cycle. The speed of CPU is known as Computer Speed
CPU SPEED MEASURES

<table>
<thead>
<tr>
<th>1 hertz or Hz</th>
<th>1 cycle per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MHz</td>
<td>1 million cycles per second or 1000 Hz</td>
</tr>
<tr>
<td>1 GHz</td>
<td>1 billion cycles per second or 1000 MHz</td>
</tr>
</tbody>
</table>

Questions for Practice

- What do you mean by a computer?
- Discuss about few input and output device of a computer.
- What are the functionalities of a computer?
- Discuss about block diagram of a computer.
- What are the characteristics of a computer?
- Write difference between RAM and ROM.
- Define the term software and hardware.
- What are the units of measurement in the computer?
- Write the comparison between system software and application software.
- What is primary and secondary memory?
Operating System

It is a control program that provides an interface between the computer hardware and the user. Part of this interface includes tools and services for the user. The Operating System is a program with the following features −

- An operating system is a program that acts as an interface between the software and the computer hardware.
- CPU allocated to which process is decided by the Operating System.
- It is an integrated set of specialized programs used to manage overall resources and operations of the computer.
- It is a specialized software that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software.

Objectives of Operating System

The objectives of the operating system are −

- To make the computer system convenient to use in an efficient manner.
- To hide the details of the hardware resources from the users.
- To provide users a convenient interface to use the computer system.
- To act as an intermediary between the hardware and its users, making it easier for the users to access and use other resources.
To manage the resources of a computer system.
To keep track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
To provide efficient and fair sharing of resources among users and programs.

Characteristics of Operating System

Here is a list of some of the most prominent characteristic features of Operating Systems –

- **Memory Management** – Keeps track of the primary memory, i.e. what part of it is in use by whom, what part is not in use, etc. and allocates the memory when a process or program requests it.

- **Processor Management** – Allocates the processor (CPU) to a process and deallocates the processor when it is no longer required.

- **Device Management** – Keeps track of all the devices. This is also called I/O controller that decides which process gets the device, when, and for how much time.

- **File Management** – Allocates and de-allocates the resources and decides who gets the resources.

- **Security** – Prevents unauthorized access to programs and data by means of passwords and other similar techniques.

**Types of Operating System:**
- Real-time
  - Multi-user vs. Single-user
- Multi-tasking vs. Single-tasking Distributed
- Embedded

**Real-Time**
- A real-time operating system is a multitasking operating system that aims at executing real-time applications.
- Responds to input instantly.

**Multi-user vs. Single-user**
- A multi-user operating system allows multiple users to access a computer system concurrently.
- Time-sharing system can be classified as multi-user systems as they enable a multiple user access to a computer through the sharing of time.
- Single-user operating systems, as opposed to a multi-user operating system, are usable by a single user at a time
Multi-tasking vs. Single-tasking
• When a single program is allowed to run at a time, the system is grouped under a single-tasking system
• While in case the operating system allows the execution of multiple tasks at one time, it is classified as a multi-tasking operating system.

Distributed
• A distributed operating system manages a group of independent computers and makes them appear to be a single computer.
• The development of networked computers that could be linked and communicate with each other, gave rise to distributed computing.

Embedded
• Embedded operating systems are designed to be used in embedded computer systems.
## Computer History

The first mechanical computer called “**Analytical Engine**” designed by **Charles Babbage** between 1830 and 1850

**Generation of Computer:**

<table>
<thead>
<tr>
<th>Generation</th>
<th>Electronic component</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Comments/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Gen. 1940-52</td>
<td>Vacuum tubes e.g. ENIAC</td>
<td>Vacuum tubes were the only electronic components available</td>
<td>Large size, generated heat, A.C. required, Unreliable constant main-tenace</td>
<td>Use machine lang. instructions. ENIAC, EDSAC, EDVAC UNIVAC, IBM-</td>
</tr>
<tr>
<td>2nd Gen. 1952-64</td>
<td>Transistors e.g. IBM 1401</td>
<td>Small size, less heat generated, more reliable</td>
<td>As above</td>
<td>Developmen t of assembly lang. IBM-162,CDC-3600, RCA-501, IBM 1401</td>
</tr>
<tr>
<td>3rd Gen. 1964-72</td>
<td>Integrated Circuits on silicon chips (IC)</td>
<td>Even smaller size, Even lower heat generation, less power</td>
<td>Initially, problems with manufacture</td>
<td>Higher level languages were used for instructions. Developmen t of software side. IBM-360 Series, ICL-1900, ICL-2900, PDP-11, HONEYWE LL-6000 Series</td>
</tr>
<tr>
<td>4th Gen. 1972-</td>
<td>Large Scale Integrated Circuits developed</td>
<td>Hardware size- very compactable, Internal</td>
<td></td>
<td>Remarkable development of software side,</td>
</tr>
<tr>
<td>5th Gen. 1984 and above</td>
<td>Very Large Scale Integrated Circuits (VLSIC)</td>
<td>Japan and many countries are working on &quot;Expert System&quot; will considerably increase man-machine interaction. Computer-aided problem solving with the help of organised information in many specialised areas and Development using ideas of artificial intelligence will be known as fifth generation.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Booting**

*Booting* is a process of starting a computer and specifically handing over the control to the operating system. There are two forms of booting, Cold Booting and Warm Booting; The notable difference between them is that the Cold Boot occurs at the beginning where the state of the primary components of the computer and RAM is checked with the help of Power On Self Test (POST) before loading the operating system. Conversely, Warm Boot skips the primary self-test and starts loading the operating system directly.

**Cold Booting**

It is the first thing that a computer does when we turn it ON from OFF position. In the very first stage, the microprocessor is initialized.

**Warm booting**

If your computer hangs due to some reason while working, and requires to be restarted to make it functional. The resetting or restarting of the computer is known as warm booting. It can be performed using the reset button or pressing the particular keys (i.e., **Ctrl+Alt+Del**) concomitantly.
Computer Viruses*
Viruses: A virus is a small piece of software that piggybacks on real programs. For example, a virus might attach itself to a program such as a spreadsheet program. Each time the spreadsheet program runs, the virus runs, too, and it has the chance to reproduce (by attaching to other programs) or wreak havoc.

• E-mail viruses: An e-mail virus travels as an attachment to e-mail messages, and usually replicates itself by automatically mailing itself to dozens of people in the victim's e-mail address book. Some e-mail viruses don't even require a double-click -- they launch when you view the infected message in the preview pane of your e-mail software [source: Johnson].

• Trojan horses: A Trojan horse is simply a computer program. The program claims to do one thing (it may claim to be a game) but instead does damage when you run it (it may erase your hard disk). Trojan horses have no way to replicate automatically.

• Worms: A worm is a small piece of software that uses computer networks and security holes to replicate itself. A copy of the worm scans the network for another machine that has a specific security hole. It copies itself to the new machine using the security hole, and then starts replicating from there, as well.

What are some tips to avoid viruses and lessen their impact?*

□ Install anti-virus software from a reputable vendor. Update it and use it regularly.

□ In addition to scanning for viruses on a regular basis, install an "on access" scanner (included in most anti-virus software packages) and configure it to start each time you start up your computer. This will protect your system by checking for viruses each time you run an executable file.

□ Use a virus scan before you open any new programs or files that may contain executable code. This includes packaged software that you buy from the store as well as any program you might download from the Internet.

□ If you are a member of an online community or chat room, be very careful about accepting files or clicking links that you find or that people send you within the community.

□ Make sure you back up your data (documents, bookmark files, important email messages, etc.) on disc so that in the event of a virus infection, you do not lose valuable work.
Language Processors: Assembler, Compiler and Interpreter

Language Processors –
Assembly language is machine dependent yet mnemonics that are being used to represent instructions in it are not directly understandable by machine and high Level language is machine independent. A computer understands instructions in machine code, i.e. in the form of 0s and 1s. It is a tedious task to write a computer program directly in machine code. The programs are written mostly in high level languages like Java, C++, Python etc. and are called source code. **These source code cannot be executed directly by the computer and must be converted into machine language to be executed.** Hence, a special translator system software is used to translate the program written in high-level language into machine code is called **Language Processor** and the program after translated into machine code (object program / object code).

The language processors can be any of the following three types:

**Compiler** –
The language processor that reads the complete source program written in high level language as a whole in one go and translates it into an equivalent program in machine language is called as a Compiler.

**Example:** C, C++, C#, Java

In a compiler, the source code is translated to object code successfully if it is free of errors. The compiler specifies the errors at the end of compilation with line numbers when there are any errors in the source code. The errors must be removed before the compiler can successfully recompile the source code again.

**Assembler** –
The Assembler is used to translate the program written in Assembly language into machine code. The source program is a input of assembler that contains assembly language instructions. The output generated by assembler is the object code or machine code understandable by the computer.

**Interpreter** –
The translation of single statement of source program into machine code is done by language processor and executes it immediately before moving on to the next line is called an interpreter. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after removal of the error. An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.

**Example:** Perl, Python and Matlab.
## Difference between Compiler and Interpreter –

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Interpreter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A compiler is a program which converts the entire source code of a programming language into executable machine code for a CPU.</td>
<td>Interpreter takes a source program and runs it line by line, translating each line as it comes to it.</td>
</tr>
<tr>
<td>Compiler takes large amount of time to analyze the entire source code but the overall execution time of the program is comparatively faster.</td>
<td>Interpreter takes less amount of time to analyze the source code but the overall execution time of the program is slower.</td>
</tr>
<tr>
<td>Compiler generates the error message only after scanning the whole program, so debugging is comparatively hard as the error can be present any where in the program.</td>
<td>Its Debugging is easier as it continues translating the program until the error is met</td>
</tr>
<tr>
<td>Generates intermediate object code.</td>
<td>No intermediate object code is generated.</td>
</tr>
<tr>
<td>Examples: C, C++, Java</td>
<td>Examples: Python, Perl</td>
</tr>
</tbody>
</table>